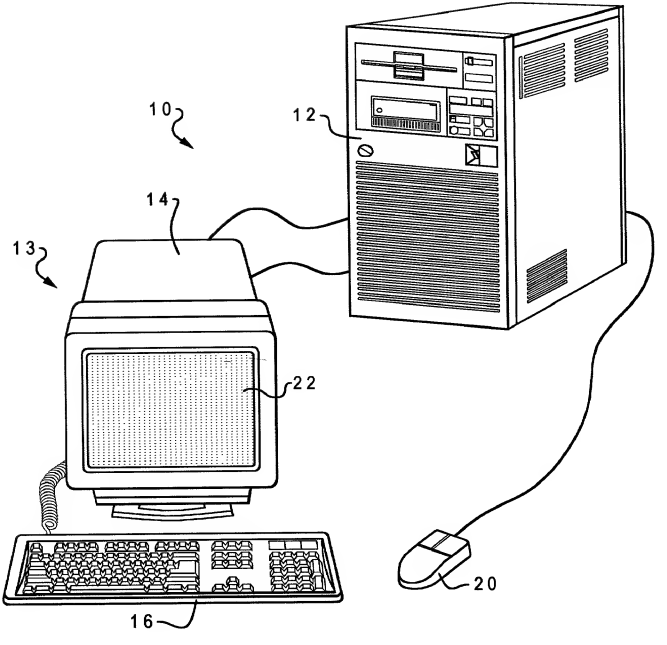

Fig. 1

10
CPU
24
ROM
30
RAM
28
DISK/TAPE DRIVE
33
I/O ADAPTER
34
COMMUNICATIONS ADAPTER
42
26
MAIN MEMORY
44
CONTROL PROGRAM
46
USER INTERFACE ADAPTER
36
DISPLAY ADAPTER
32
16
14
40
20
38

Fig. 2

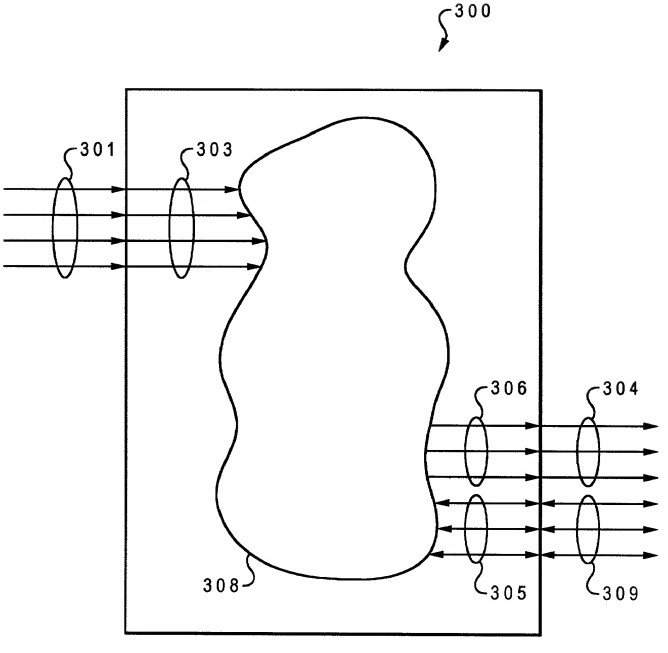

Fig. 3A

Fig. 3B

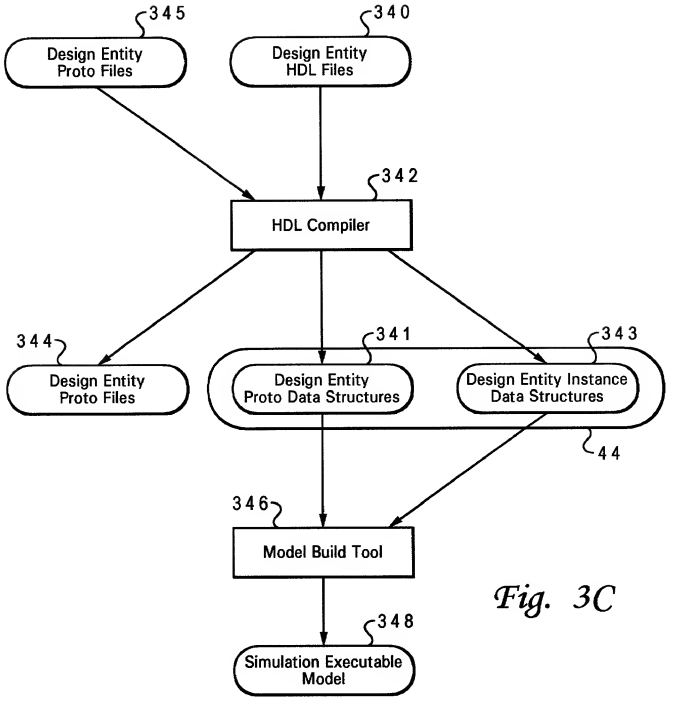

Fig. 3C

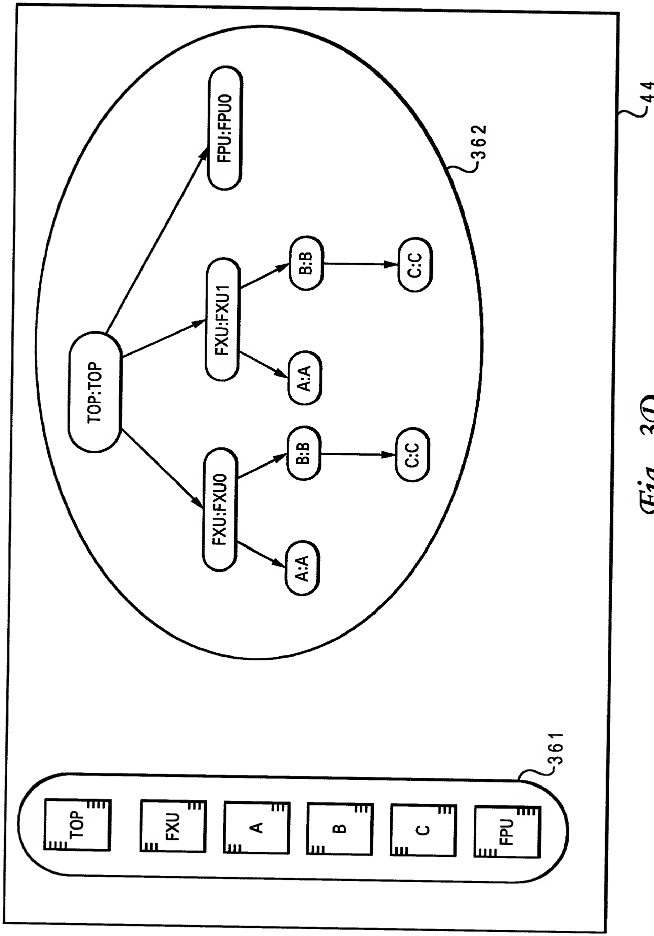
TOP:TOP
FXU:FXU0
FXU:FXU1
FPU:FPU0
A:A
B:B
C:C
A:A
B:B
C:C
TOP
FXU
A
B
C
FPU
361
362
44

Fig. 3D

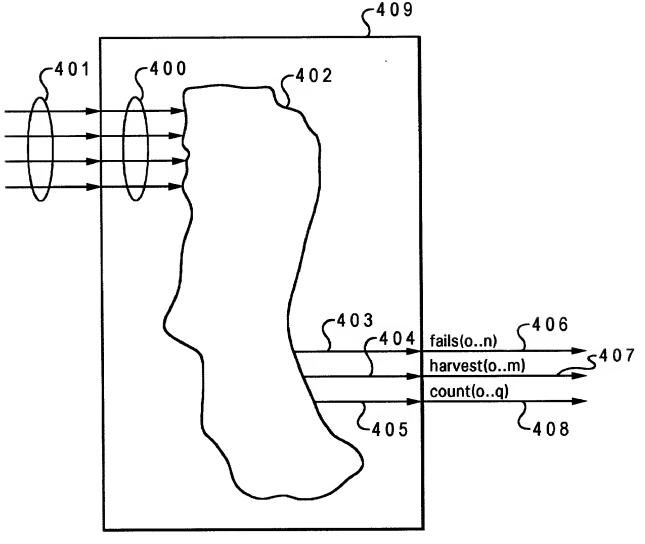
409
401
400
402
403
404
406
fails(o..n)
407
harvest(o..m)
count(o..q)
405
408

Fig. 4A

TOP:TOP
FXU:FXU0(a)
A:A
Q
372
B:B
C:C
S
374
FXUCHK
R
376
321b
325a
326a
327a
410
321a
415
FXU:FXU0(b)
A:A
Q
B:B
C:C
S
FXUCHK
R
325b
326b
327b
411
416
FPU:FPU0
FPUCHK
412
322
417
420
329
Counters
421
422
Harvest Flag
423
Fail
424

Fig. 4B

```
ENTITY FXUCHK IS

    PORT(   S_IN        :   IN std_ulogic;
            Q_IN        :   IN std_ulogic;
            R_IN        :   IN std_ulogic;
            clock       :   IN std_ulogic;
            fails       :   OUT std_ulogic_vector(0 to 1);
            counts      :   OUT std_ulogic_vector(0 to 2);
            harvests    :   OUT std_ulogic_vector(0 to 1);
        );

452 --!! BEGIN
    --!! Design Entity: FXU;

    --!! Inputs
453 --!! S_IN       =>    B.C.S;
    --!! Q_IN       =>    A.Q;
    --!! R_IN       =>    R;
    --!! CLOCK      =>    clock;
    --!! End Inputs

    --!! Fail Outputs;
454 --!! 0 : "Fail message for failure event 0";
    --!! 1 : "Fail message for failure event 1";
    --!! End Fail Outputs;

    --!! Count Outputs;
455 --!! 0 : <event0> clock;
    --!! 1 : <event1> clock;
    --!! 2 : <event2> clock;
    --!! End Count Outputs;

    --!! Harvest Outputs;
456 --!! 0 : "Message for harvest event 0";
    --!! 1 : "Message for harvest event 1";
    --!! End Harvest Outputs;

457 --!! End;

    ARCHITECTURE example of FXUCHK IS

    BEGIN

        ... HDL code for entity body section ...

    END;
```

450

451

458

440

Fig. 4C

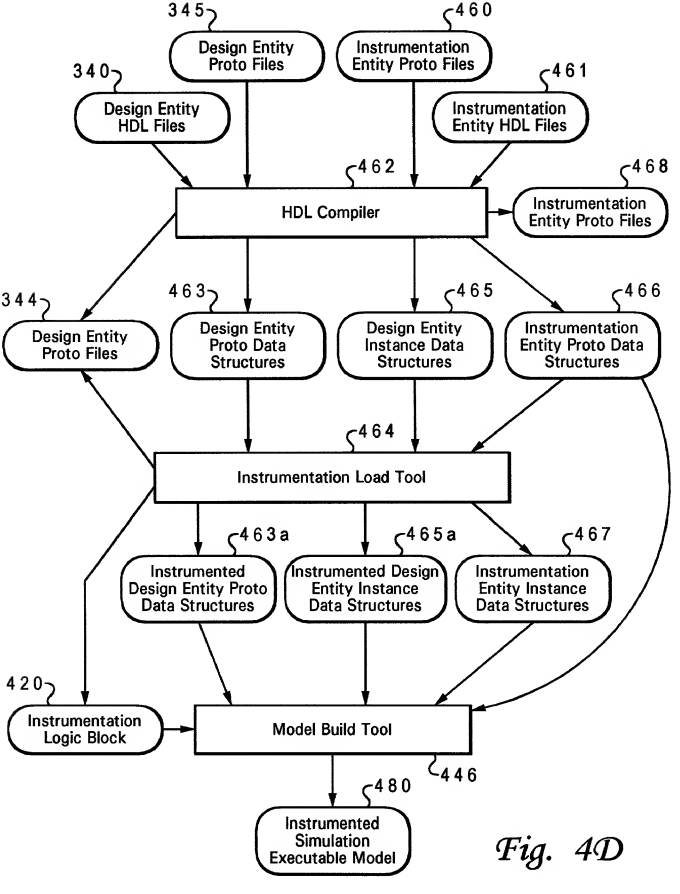

Fig. 4D

TOP
FXU
A
B
C
FPU
FXUCHK
FPUCHK
481
TOP:TOP
FXU:FXU0
FXU:FXU1
FPU:FPU0
A:A
FXUCHK
B:B
C:C
A:A
FXUCHK
B:B
C:C
FPUCHK
482
44

Fig. 4E

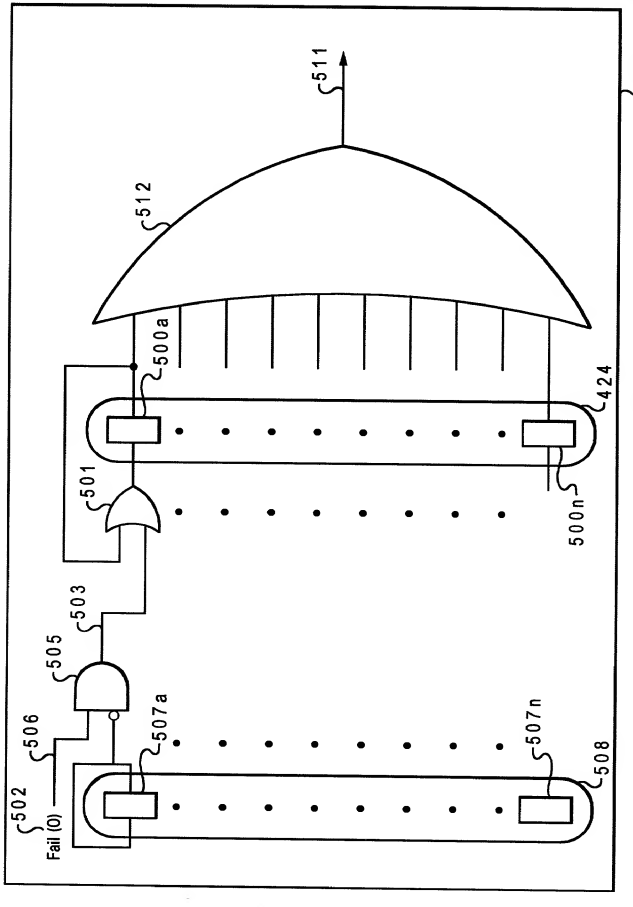
502
Fail (0)
506
505
503
501
507a
507n
508
500a
500n
424
512
511
420

Fig. 5A

Fig. 5B

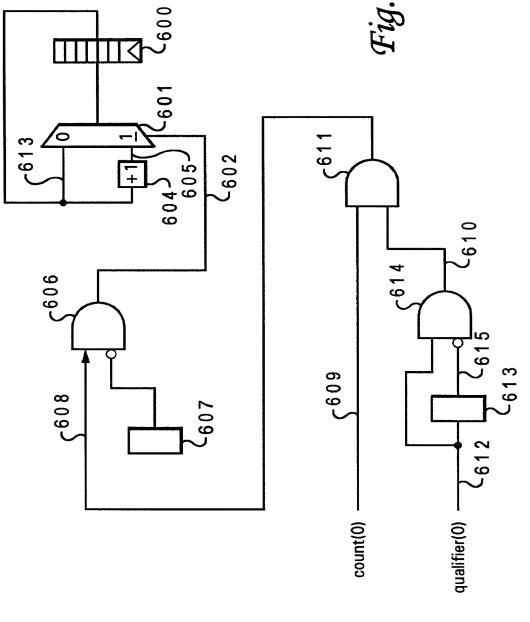

Fig. 6A

Fig. 6B

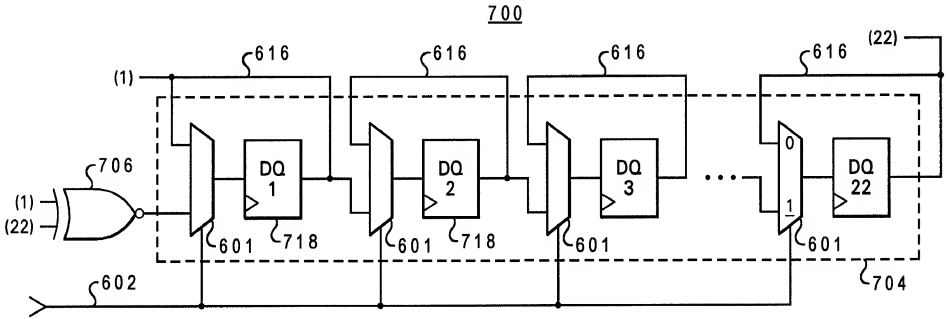
700
616
616
616
616
(22)
(1)
706
(1)
(22)
DQ 1
DQ 2
DQ 3
DQ 22
0
1
601
718
601
718
601
601
602
704

Fig. 7

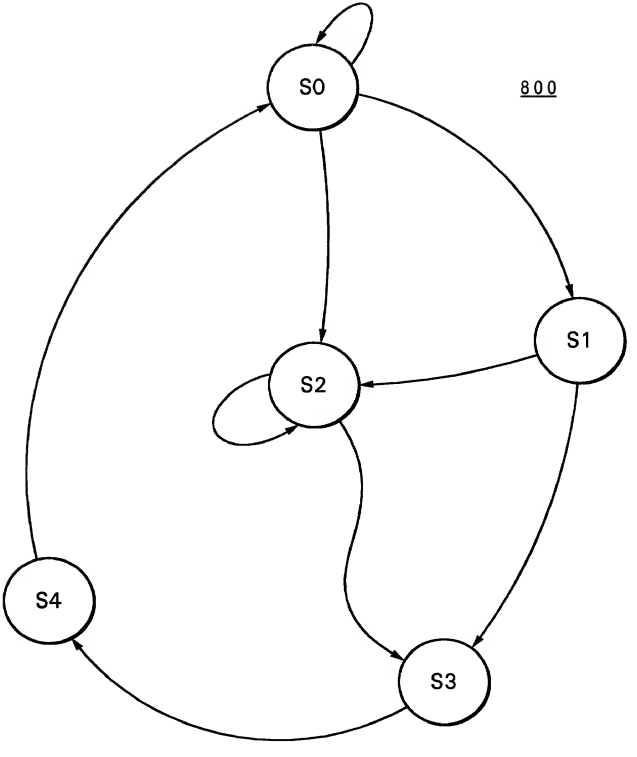

Fig. 8A
Prior Art

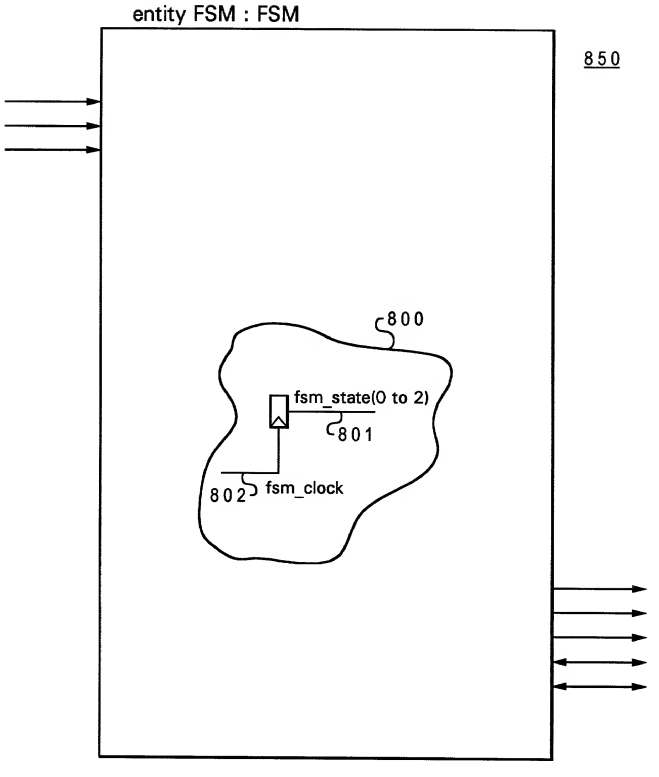

Fig. 8B
Prior Art

```
ENTITY FSM IS

    PORT(
            ....ports for entity fsm....
        );

    ARCHITECTURE FSM OF FSM IS

    BEGIN

            ... HDL code for FSM and rest of the entity ...

            fsm_state(0 to 2) < = ... Signal 801 ...

 853 {  --!! Embedded FSM : examplefsm;
 859 {  --!! clock         : (fsm_clock);
 854 {  --!! state_vector  : (fsm_state(0 to 2));
 855 {  --!! states        : (S0, S1, S2, S3, S4);
 856 {  --!! state_encoding : ('000', '001', '010', '011', '100');
        --!! arcs          : (S0 = > S0, S0 = > S1, S0 = > S2,
 857 {  --!!                  (S1 = > S2, S1 = > S3, S2 = > S2,
        --!!                  (S2 = > S3, S3 = > S4, S4 = > S0);
 858 {  --!! End FSM;

                    ,
                    ,
                    ,
                    ,
                    ,
                    ,
                    ,

    END;
```

852 } 860

Fig. 8C

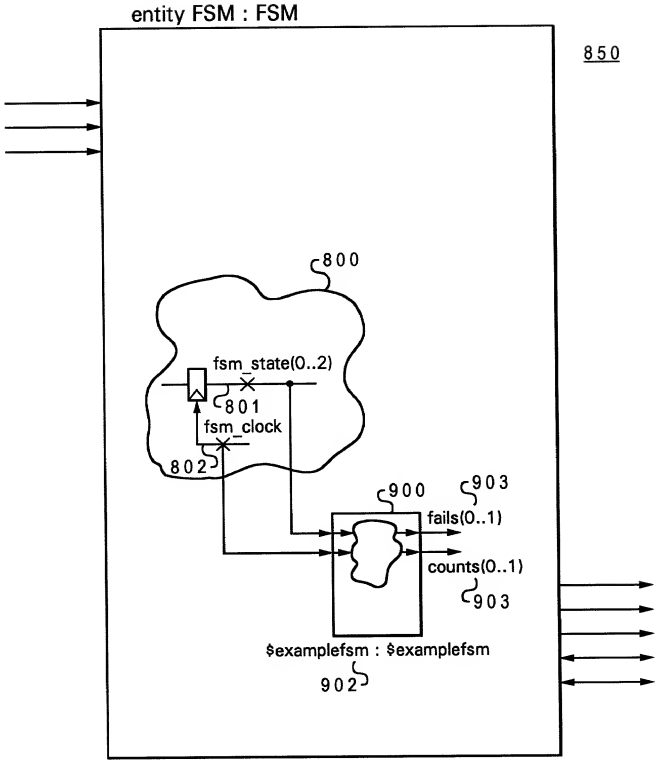
entity FSM : FSM
850
800
fsm_state(0..2)
801
fsm_clock
802
900
903
fails(0..1)
counts(0..1)
903
$examplefsm : $examplefsm
902

Fig. 9

Fig. 10A

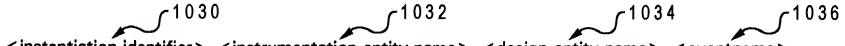

Fig. 10B

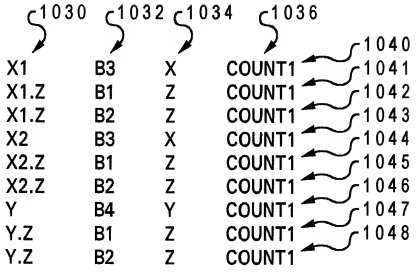

Fig. 10C

Fig. 10D

Fig. 11A

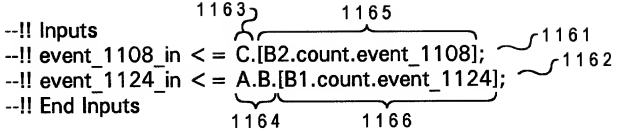

Fig. 11B

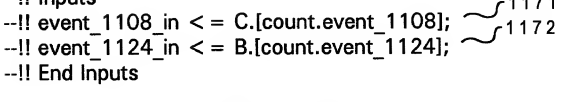

Fig. 11C

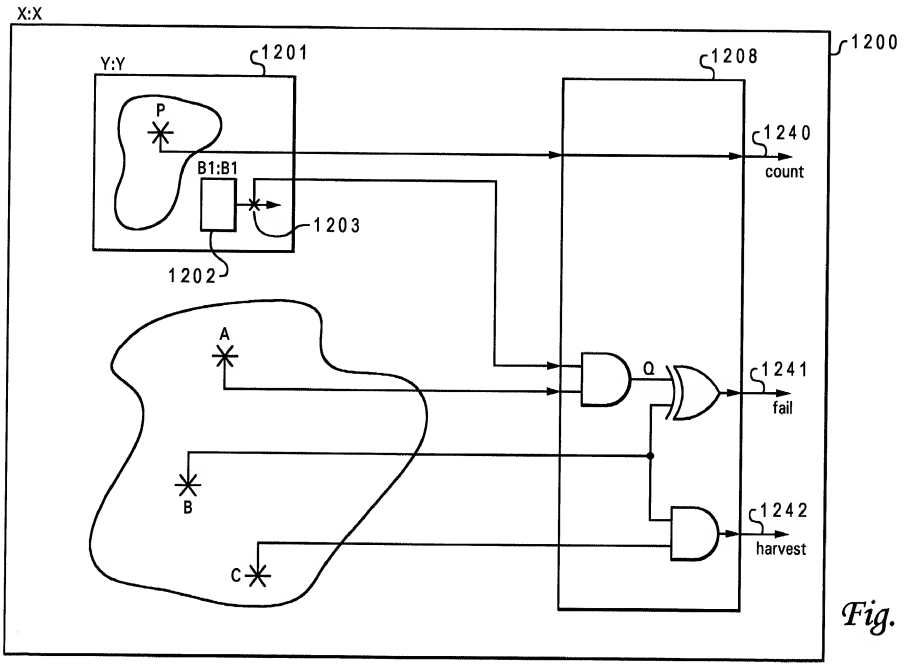
X:X
Y:Y
1201
1208
1200
P
1240
count
B1:B1
1203
1202
A
Q
1241
fail
B
1242
harvest
C

Fig. 12A

```
ENTITY X IS
    PORT(    :
             :
             :
         );

ARCHITECTURE example of X IS

BEGIN
    .
    .
    .
    .
    .
  ... HDL code for X ...
    .
    .
    .
    .
    .
    Y:Y
    PORT MAP(    :
                 :
             );

    A <= ....
    B <= ....
    C <= ....

    --!! [count, countname0, clock] <= Y.P;
    --!! Q <= Y. [B1.count.count1] AND A;
    --!! [fail, failname0, "fail msg"] <= Q XOR B;
    --!! [harvest, harvestname0, "harvest msg"] <= B AND C;

END;
```

1220 1221 1222 1223 1230 1232 1234 1236

Fig. 12B

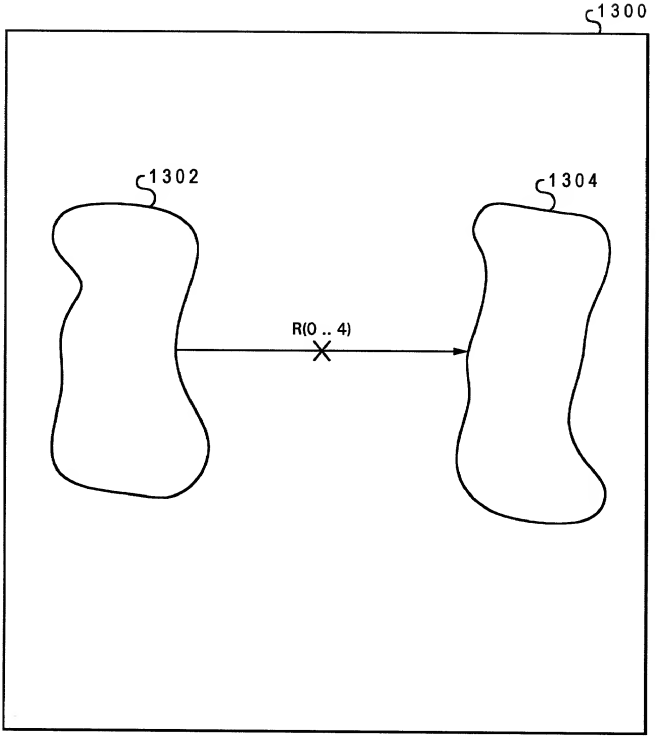

Fig. 13A

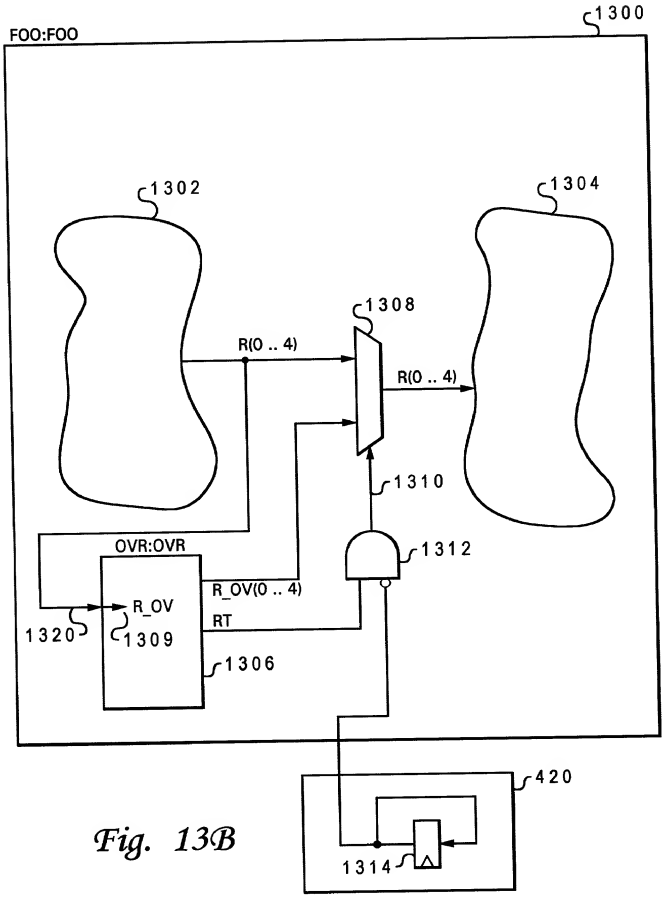
1300
FOO:FOO
1302
1304
1308
R(0 .. 4)
R(0 .. 4)
1310
1312
OVR:OVR
R_OV(0 .. 4)
R_OV
RT
1320
1309
1306
420
1314

Fig. 13B

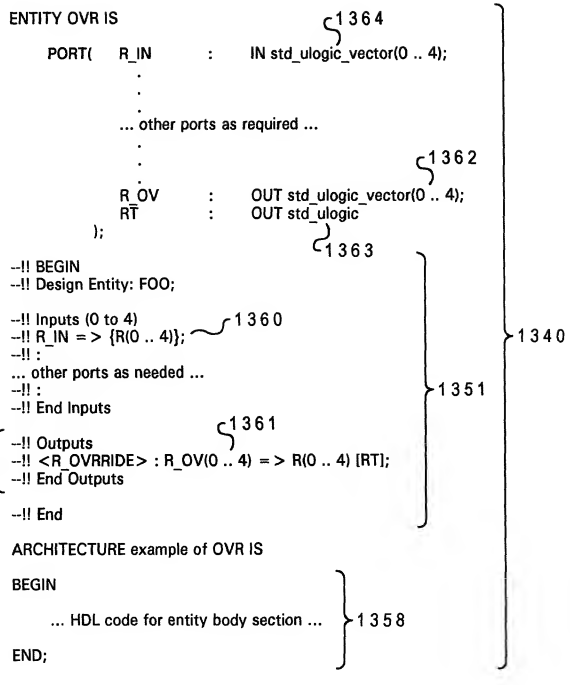
ENTITY OVR IS
1364
PORT(R_IN : IN std_ulogic_vector(0 .. 4);
... other ports as required ...
1362
R_OV : OUT std_ulogic_vector(0 .. 4);
RT : OUT std_ulogic
);
1363
--!! BEGIN
--!! Design Entity: FOO;
--!! Inputs (0 to 4)
1360
--!! R_IN => {R(0 .. 4)};
--!! :
... other ports as needed ...
--!! :
--!! End Inputs
1361
1356
--!! Outputs
--!! <R_OVRRIDE> : R_OV(0 .. 4) => R(0 .. 4) [RT];
--!! End Outputs
--!! End
ARCHITECTURE example of OVR IS
BEGIN
... HDL code for entity body section ...
END;
1340
1351
1358

Fig. 13C

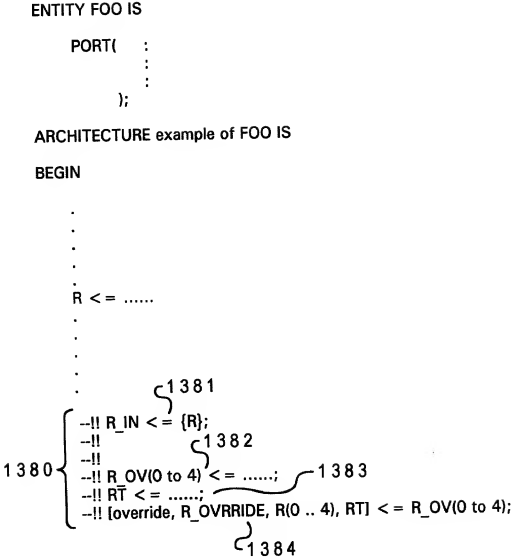
ENTITY FOO IS
PORT(:
:
:
);
ARCHITECTURE example of FOO IS
BEGIN
R <= .....
1381
1380
--!! R_IN <= {R};
--!!
1382
--!!
--!! R_OV(0 to 4) <= .....;
1383
--!! RT <= .....;
--!! [override, R_OVRRIDE, R(0 .. 4), RT] <= R_OV(0 to 4);
1384

Fig. 13D